Semiannual Status Report
(NAG 5–916)

Development and Evaluation of Packet Video Schemes

Submitted to

Instrument Division
Engineering Directorate
Goddard Space Flight Center
Greenbelt, MD 20771
Attn: Warner Miller and Pen Shu Yeh

K. Sayood, Y.C. Chen, and A. C. Hadenfeldt
Department of Electrical Engineering
Center for Communication and Information Science
University of Nebraska–Lincoln

Period June 1990 — December 1990

INTRODUCTION

Reflecting the two tasks proposed for the current year, namely a feasibility study of simulating the NASA network, and a study of progressive transmission schemes, the report is divided into two main sections. The first section provides our view of the NASA network, gleaned from the various technical reports made available to use. Also included is a brief overview of how the current simulator could be modified to accomplish the goal of simulating the NASA network. As the material in this section would be the basis for the actual simulation, it is important to make sure that it is an accurate reflection of the requirements on the simulator.

The next section contains brief descriptions of the set of progressive transmission algorithms selected for the study. The results available in the literature have been obtained under a variety of different assumptions, not all of which are stated. As such, the only way to compare the efficiency and the implementational complexity of the various algorithms is to simulate them. This is our goal during this period.

NETWORK SIMULATION

In this section we describe our understanding of the NASA network and the various aspects that need to be simulated.

Environment

The Advanced Orbiting System will need to support a variety of users. Basically, two different sets of payload users will use the communication facilities of the NASA network. One set of users fits into the category "Experimental user," while the other set of users belongs to the category "Observational user." Beside these, the functions of system operation and maintenance will also need some interactive communication capabilities to support the payload users. Different users may have different transmission requirements which reflect differences in the mode of operation. However, basic communication resources must be shared by all users. Activities of various users may differ in the duration of communication, the level of interaction that is required in transmission, and the volume of data that needs to be transmitted to the ground.

Consider the experimental user who might be conducting experiments in materials processing or life-sciences. Examples of such experiments include the

development of ultra-pure material, the development of techniques for building space structures in pressurized space vehicles, and the testing of human ability to adapt to the space environment. Experiments like these are generally operated for a limited time, typically just a few hours, and a crew member may monitor the progress of an experiment from workstations at different locations. Hence, source-destination connection will typically exist only for relatively short sessions. The level of human interaction is high, as much of the data that is generated from the experiment may be evaluated onboard by a crew member who is monitoring the experiment. Therefore, the volume of data that needs to be transmitted to ground station is relatively low.

The observational user deals with astronomy, investigation of space physics phenomena, and monitoring the environment on the Earth. Instruments located on platforms collect and transmit raw data to the ground for detailed analysis. The volume of raw data is huge since the instrument is turned on for a long time during the flight and the instrument usually operates at a high data rate. In most cases, the ground processing facility will remain fixed for months or even years, thus the association between a space instrument and its ground workstation is stable.

Finally, for the interactive user, like those involved in network management, the requirements fit somewhere between those of the Experimental and Observational users. Most of the time, reliability is of most concern and therefore a very robust and carefully validated suite of communications protocols is required.

The space mission environment presents unique problems for networking. One unavoidable problem is the very large one-way propagation delay (often in the order of one half second). This large delay, the very weak signal levels, the presence of noise, and the very high doppler shifts caused by relative vehicle motion, must be taken into consideration when designing reliable protocols for space links. Other problems involve the high cost and relative scarcity of data communications resources. The intermittency of space-to-ground communications imposes unique constraints since most individual space vehicles have been scheduled for only short contact periods per orbit (usually 10 to 20 minutes). Accordingly, in order to provide complete and continuous data sets to support user analysis, most of the data from the flight systems must be stored onboard during frequent link outages, for transmission during the next period of contact. Thus stored data will overlap somewhat with real-time data. Meanwhile, since technology and cost may dictate that data be stored by recording them on tape, and reliability and operational simplicity requirements may rule against rewinding the tapes before playback, the replayed data is often transmitted in reverse order. A specific function called "Level Zero Processing" (LZP) is performed at the receiving station. LZP includes reversal of taped data, removal of overlaps between the stored and the real-time data, and restoration of data to their as-generated sequential order.

CCSDS Principal Network

A "CCSDS Principal Network" (CPN) serves as the project data handling network which provides end-to-end data flow in support of the Experimental, Observational and interactive users of the Advanced Orbiting System. A CPN consists of an "Onboard Network" in an orbiting segment connected through a CCSDS "Space Link Subnetwork" (SLS) either to a "Ground Network" or to another Onboard Network in another orbiting segment. The SLS is the central component of a CPN: it is unique to the space mission environment and provides customized services and data communications protocols. Within the SLS, CCSDS defines a full protocol to achieve "cross support" between agencies. Cross support is defined as the capability for one space agency to bidirectionally transfer another agency's data between ground and space systems using its own transmission resources. A key feature of this protocol is the concept of "Virtual Channel" which allows one physical space channel to be shared among several data streams, each of them may have different service requirements. A single Physical space channel may therefore be divided into several logical data channels, each known as a Virtual Channel.

Eight separate services are provided within a CPN. Two of these services ("Path" and "Internet") operate end-to-end across the entire CPN. They are complementary services, which satisfy different user data communications require-

ments: some users will interface with only one of them, but many will operate with both. The remaining six services ("Encapsulation," "Multiplexing," "Bitstream," "Insert," "Virtual Channel Data Unit" and "Physical Channel") are provided only within the Space Link Subnetwork for special applications such as audio, video, highrate payloads, tape playback, and the intermediate transfer of Path and Internet data.

End-to-End Services

The Path service serves primarily to route high volume traffic (mostly from Observational users) across the network. In this sense, it serves as a network layer. No protocol driven transport, session and presentation layers are required because they are pre-established by management. So the Path service interfaces above to the Application layer directly and interfaces below to the Datalink layer. The data unit for Path service is the primary CCSDS Packet. The objectives of the Path service are both to minimize onboard processing, because of stringent constraints on onboard weight, power and volume, and to optimize the utilization of space channel bandwidth. Since the source and destination pairs in the Path service are quite stable, network management system preconfigures a Logical Data Path (LDP) which connects source and destination. Data is then routed across the CCSDS Principal Network by specifying only the LDP to be followed, rather than complete source and destination addresses. In this way the Path

service creates many "trunks" for efficient transmission of large volumes of data. The LDP is represented by Application Identifier (APID) which is 11 bits long and can represent up to 2048 global data paths. If the 2048 global assignments are insufficient, then the APID qualifier which is a field of the Virtual Channel Data Unit (VCDU) is used to identify the LDP owner entity (typically a complete spacecraft module). Simplicity is the key to both the architecture and the functionality of the CCSDS Path service. Once a LDP has been configured, large volumes of data can be efficiently transmitted across the path. The CCSDS Packet contains a Primary Header, an optional Secondary Header, and a variable-length User Data Field. The Primary Header, which is 6-octets long, contains information to identify the LDP, to delimit the Packet boundaries by specifying its length, and to control Packet sequencing. Use of such a simple Packet structure reduces the functionality necessary at the interface between the instrument and the onboard subnet, minimizes the amount of processing required to transmit a Packet across the CPN, and optimizes use of bandwidth over the various subnets.

The other end-to-end service is the Internet service, which supports the Experimental users and the interactive communication needs, using standard OSI-compatible services and protocols. The Internet service layer maps directly into the OSI protocol stack, interfacing to the Data Link layer below and the Transport layer above. The type of Internet service data differs from Path service data in two major ways:

1. The number and location of the application endpoints may change quite frequently.
2. The data rate is relatively low and the duty cycle of individual users is more intermittent.

Therefore, for Internet service, channel bandwidth and processing efficiencies are not so important. Instead, flexibility is the first consideration. In order to achieve flexibility, CCSDS adopted an existing commercially-supported protocol, the ISO 8473 Connectionless Network Protocol Specification, to support the Internet service. The ISO 8473 Packet is the Internet service data unit. The Internet Packet header is much more complex than the CCSDS Packet header, making it more flexible but requiring additional processing time. Since the Internet packet header specifies full global endpoints addresses, so Internet routing is straightforward and follows OSI techniques. In order to reduce operational complexity, the only packet structure which is recognized within the Virtual Channels on the Space Link Subnetwork is the CCSDS Packet. Because of this, the Internet Packets must be encapsulated within the CCSDS Packets during transmission through the Space Link Subnet. These encapsulated packets are multiplexed with other Packets within the Coded Virtual Channel Data Unit (CVCDU). The coding is required in order to provide sufficient error protection so that the packet headers are not corrupted during transfer. One CCSDS Application Process ID is globally reserved so that the encapsulated Internet packets can be

identified via the CCSDS Packet Header. When CVCDUs containing Internet data reach the ground, the unique CCSDS Packet APID associated with them is recognized and the CCSDS Packet Header is stripped. The reconstituted ISO 8473 packets are then forwarded directly via ISO networks to the end user.

Space Link Subnet Services

The Space Link Subnet supports the bidirectional transmission of data through the space/ground and space/space channels which connect the distributed elements of the CCSDS Advanced Orbiting Systems. It also provides "direct connect" transmission services for certain types of data which require timely or high-rate access to the space channel. During SLS transfer, different flows of data are separated into different Virtual Channels, based on data handling requirements at the destination. These Virtual Channels are interleaved onto the physical channel as a serial symbol stream. A particular Virtual Channel may contain either packetized or bitstream data, or a combination of both.

The Space link Subnetwork consists of two layers: the Space Link layer and the Space Channel layer which correspond to the ISO-equivalent Data Link layer and Physical layer respectively. Efficient use of the physical space channel was a primary driver in the development of these protocols. The Space Link layer can be further decomposed into the Virtual Channel Link Control sublayer (VCLC) and the Virtual Channel Access sublayer (VCA).

The main function of the VCLC sublayer is to convert incoming data into a protocol data unit which is suitable for transmission over the physical space channel. Four type of protocol data units may be generated by the VCLC sublayer: fixed length blocks of CCSDS Packets, called "Multiplexing Protocol Data Units" (M-PDUs); fixed length blocks of Bitstream data, called "Bitstream Protocol Data Units" (B-PDUs); fixed length blocks of mixed packetized and isochronous data, called "Insert Protocol Data Units" (IN-PDUs); and fixed length blocks of data for use by retransmission control procedures, called "Space Link ARQ Procedure Protocol Data Units" (SLAP-PDUs).

The VCLC sublayer contains several procedures to perform its various functions. The Encapsulation procedure provides the flexibility to handle virtually any packet structure. It puts a primary header to delimited data units (including Internet packet) to convert them into a CCSDS Packet. The multiplexing procedure multiplexes the CCSDS Packets on the same virtual channel together. The length of multiplexing protocol data unit is fixed since it is required to fit exactly in the fixed length data space of VCDU/CVCDU. Therefore, there may be some packets which are contained in two or more M-PDU. In this case the "first packet pointer" which points out where the first packet starts is essential. Some user data, such as audio, video, playback and encrypted information, will simply be presented to the SLS as a stream of bits or octets. The Bitstream procedure simply blocks these data into individual Virtual Channels and transmits it. When the

transmission rate is high, Bitstream data may be transmitted in dedicated Virtual Channel. Alternatively, if the transmission rate is low, it can be inserted at the front of other packetized or bitstream data. This is called the Insert procedure.

The last procedure available to the VCLC is the Space Link ARQ Procedure (SLAP) which is used to provide guaranteed Grade-1 delivery on data links that connect the space and ground elements of a CPN. The SLAP incorporates the following features:

1. It provides delivery of a single stream (in each direction) of user data across a space/ground or space/space link.
2. It provides delivery of the user data in the order received, without omissions or duplication.
3. It provides a full duplex services using a pair of Virtual Channels dedicated to Grade-1 services. Once a connection is established, data transfers on the forward and return links are asynchronous with respect to each other.
4. It provides for automatic recovery from routine space link transmission errors.
5. It provides for automatic reestablishment of the link connection after an interruption, with notification to the user.
6. It provides the means to acknowledge one or more transmissions in a single report.
7. It provides the means to test the space link before sending data.

8. It provides the means to request the status of a remote receiver.
9. It provides the opportunity for a receiver to report status to a sender at some minimum period.
10. It conserves link bandwidth, including the original transmission, retransmission and transfer of supervisory and reporting data.
11. It is designed to support transfer rates of 1 to 100 Mbps.
12. It can accommodate a loop time in the range of 0.5 to 2.0 seconds. This assumes use of one, or possibly two, data relay satellites, to communicate between ground and space or space to space.
13. It is expandable to accommodate selective retransmission for use with higher data rates or for long-delay applications.

SLAP-PDU carries "Link ARQ Control Words" (LACWs) which report progress on receipt of data flowing in the opposite direction. Upon arrival at the receiving end, the LACW is extracted from the PDU, and the sequence number is checked to assure that no data has been lost or duplicated. In the event of a sequence error, the LACW carried by PDUs traveling in the opposite direction is used to signal that a retransmission is required. This retransmission begins with the first PDU that was not received in sequence, and all subsequent PDUs are retransmitted in the order in which they were originally provided to the LSAP from the layer above.

The VCA sublayer creates the protocol data units used for space link data transfer: These are either "Virtual Channel Data Units" (VCDUs) or "Coded Virtual Channel Data Units" (CVCDUs), and are formed by appending fixed length Header, Trailer and (for CVCDUs) error correction fields to the fixed length data units generated by the VCLC sublayer. The VCA sublayer is composed of the Virtual Channel Access (VCA) and the Physical Channel Access (PCA) procedures. VCA procedure generates VCDU for protocol data units which come from VCLC sublayer or accepts independently generated VCDU from reliable users. A VCDU with a powerful outer code of error-correcting Reed-Solomon check symbols appended to it is called a CVCDU. Relative to a VCDU, a CVCDU contains more error-control information and, hence, less user data. The "Virtual Channel ID" which is the field of the VCDU/CVCDU Header can enable up to 64 virtual channels to be established concurrently for each assigned Spacecraft ID on a particular physical space channel. Since space data is transmitted through weak signal over a noisy channel as a serial symbol stream, a robust frame synchronization process at the receiving end is required. Therefore, fixed length VCDU/CVCDU is used and the PCA procedure prefixes a 32 bits Synchronization Marker in front of VCDU/CVCDU to form a "Channel Access Data Unit" (CADU). A contiguous and continuous stream of fixed length CADUs, known as a "Physical Channel Access Protocol Data Unit" (PCA-PDU) is transmitted as individual channel symbols through the ISO-equivalent Physical Layer of the

Space Link Subnet, which is known as the "Space Channel Layer."

Space Link Grades Of Service

Three different "grades of services" are provided by the Space Link Subnet, using a combination of error detection, error correction and retransmission control techniques. However, each virtual channel can only support a single grade of service.

(1) Grade-3 Service

This service provides the lowest quality of service. Data transmitted using Grade-3 service may be incomplete and there is a moderate probability that errors induced by the Space Link Subnet are present and that the sequence of data units is not preserved. A VCUD is discarded if an uncorrectable error is detected at the destination. Grade-3 service should not be used for transmission of asynchronous packetized data, because it provides insufficient protection for the extensive control information contained in the packet headers.

(2) Grade-2 Service

CVCDU is the unit of transmission that support Grade-2 service. The Reed-Solomon encoding provides extremely powerful error correction capabilities. Data transmitted using Grade-2 service may be incomplete, but data sequencing is preserved and there is a very high probability that no data errors have been induced by the Space Link Subnet.

(3) Grade-1 Service

Data transmitted using Grade-1 service is delivered through the Space Link Subnet complete, in sequence, without duplication and with a very high probability of containing no errors. This grade of transmission is provided by using two paired Reed-Solomon encoded Virtual Channels, in opposite directions, so that an Automatic Repeat Queueing (ARQ) retransmission scheme may be implemented.

Network Simulator

The network simulator will be developed based on an existing Prairielink Simulator. Some major parts of the simulator are described here.

The PLTOPOLOGY program is used to generate a topological description of the network to be simulated. It contains the number of nodes, the definition (includes connectivity and propagation delay) of the links between nodes, and the initial bit error rate for each link. A graphic mode has been included into the topology program to create graphic display maps for use in the simulator.

The PLTRAFFIC program is used to generate a file containing the cumulative distribution functions (cdf's), session types and other traffic data required by the simulator. The philosophy for setting up traffic on the network and for the simulated generation of messages is to set up a variety of "session types" which will be used to define the statistics of the various session parameters and then to randomly generate the types of session expected to exist for each node pair in the

network on a source to destination basis. The program PLSIMPREP is used to generate a checkpoint file which contains all the data needed for the simulation including the topology, traffic and network parameters for various network layers.

PLSIMEX is the generic name for the simulator. The system has been assembled in modular form so that functional units can be replaced with modules using different protocols if other algorithms are to be tested. The three principal programs which perform the function of the network layer are: PLSESSION, PLNETWORK, PLDL. All layers above the Network layer are combined into one Session layer which allows users to establish or terminate "Sessions" by tasks such as "SL_Connect_Request," "SL_Connect_Confirm," "SL_Disconnect_Initiate." At message arrival time, the Session layer generates message with all of its randomly selected attributes according to the checkpoint file. With cdf parameters chosen carefully, we can generate the Path, Internet and other traffic type data in the Advanced Orbiting System. The Network layer is concerned with controlling the operation of the network. A key design issue is to determine how packets are routed from source to destination. Another issue is how to avoid the congestion caused if too many packets are presented into the network at the same time. In the simulator, the Network layer performs all the functions related to these two aspects including dialing up new virtual channel when more capacity is required and releasing them when not needed ("NL_Connect_Initiate (Accept, Comfirm..)", "Release_VC"), network processing and queue handling,

routing and flow control ("NL_Flow_Control," "Network_Processor"). In order to provide Path and Internet services in the Network layer, packet construction procedure is needed when the traffic type is an octet string. Two service data units (CCSDS Packet and Internet Packet) need to be generated with different packet header. There is one module for the routines which are common to most routing algorithms which can be applied for Internet Packet. On the other hand, setting up the Logical Data Path for the CCSDS packets requires a different algorithm to assign virtual channels. The Datalink layer is simulated using the DCA Inner Protocol (DCAIP) in conjunction with the DEC's DDCMP datalink protocol. In structure, it is very much like the two sublayers (VCLC and VCA) in the Space Link Layer. But some modules have to be built in the simulator to perform Encapsulation, Multiplexing, Insert, Bitstream and SLAP procedures. Also, a large number of packet types need to be defined in the environment. Error protection and synchronization schemes have not been dealt with in the existing simulator but can be adopted in the future.

PROGRESSIVE TRANSMISSION

In this section we present a brief description of the progressive transmission schemes under study. We are currently in the process of simulating some of these schemes in order to evaluate performance and complexity issues.

Progressive transmission schemes may be divided into two categories: those that use transform coding, and those that do not. Transform-based coders are usually lossy coders, while other types of coders are usually lossless. However, it is possible to make a lossless coder into a lossy one by early termination of the algorithm, and it is also possible to make a lossy coder into a lossless one by adding some type of lossless residual error coder as a final step.

For the schemes that use some form of transform coding, the discrete cosine transform (DCT) seems to be the most popular. The DCT does a good job of compacting energy into only a few components; thus, a good representation of an image may be obtained by retaining only a few coefficients of the transform. Also, a progressive transmission scheme is easy to implement by transmitting only a few of the DCT coefficients at a time. In addition, fast algorithms for computing the DCT exist, and hardware processors which perform this operation are available on a single chip.

For schemes that do not use some sort of transform coding, there are many variations. Nearly the only thing they have in common is the ability to generate coded data in progressive stages which can be used to reconstruct the source image (possibly with some distortion, which hopefully is not too apparent). Since most of the non-transform schemes are lossless coders, these coders typically require higher channel data rates than the lossy (e.g., transform) coders.

Some of the schemes attempt to improve the perceptual quality of early passes by compensation based on a model of the human visual system. The compensation is typically aimed at reducing the bit rate while introducing distortion in ways which the eye cannot detect easily.

Comments About Implementation

As a general rule, transform-based schemes tend to be more complex from an implementation standpoint. These schemes tend to require more storage space for both encoder and decoder, and usually involve many floating-point calculations. Non-transform schemes may also require a significant amount of floating-point computation, but often involve only a simple difference or averaging operation. The availability of digital signal processing (DSP) chips and cheap memory devices may lessen the significance of these problems.

Since many of the schemes divide the source image into blocks (e.g, 8x8 or 16x16), some form of parallel processing should be possible. This issue, however,

has not been discussed in detail by any of the authors. It is also possible that some of the schemes could be reworked slightly to a more parallel form.

Transform-Based Schemes

Ngan [1984] discusses five methods of transmitting the coefficients of an 8x8 DCT block from the perspective of achieving the best perceptual quality with the fewest number of transmitted coefficients. The scheme found to work best uses a distortion measure to determine the optimal transmission sequence for the AC coefficients. Coefficients are ranked according to the mean-squared error of the received image when only that coefficient is transmitted together with the DC coefficient. While this method performs better than the others, it requires additional system overhead which the remaining four methods do not. Of these, a "zig-zag" transmission sequence performs best. This sequence sends the low spatial frequency DCT coefficients first, and proceeds by sending higher and higher frequency coefficients. The described research is not considered to be a complete progressive transmission scheme, since implementation issues such as coefficient quantization and bit allocation strategies are not discussed. However, the zig-zag transmission sequence is used as a part of some of the other complete schemes below.

Dubois and Moncet [1986] coded NTSC color images using a 16x16 DCT in several forms. One scheme, which was designed for progressive transmission,

sorted the DCT coefficients into groups according to their properties and energies. These groups were then uniformly quantized and progressively transmitted using a Huffman code specifically designed for each group. Also included in the Huffman codebook were symbols representing run-lengths of zero coefficients, since there are usually many small amplitude coefficients in a typical block. In this scheme, the quantizer step size is used to control the coder data rate (at the expense of increased coefficient errors). A variation of the scheme sends each coefficient group using the same Huffman code. Both schemes gave approximately the same performance, although the latter is less complex. In a third variation of the scheme, a lower bit rate was achieved by taking advantage of the properties of the human visual system. In this case, high-frequency transform coefficients were quantized with a larger quantizer step size, on the assumption that at least some of this increased high frequency noise (error) would be filtered out by the eye. For this last scheme, a reduction in coding rate of 24-33 percent over the first two schemes is claimed, without a significant increase in visible distortion. The system is of medium complexity due to the cosine transform, but otherwise should not be difficult to implement. The amount of overhead information needed for the system is small. It should also be possible to improve the performance of this system by more carefully encoding the DCT coefficients (e.g., nonuniform quantization, simple vector quantization) without greatly increasing this complexity.

Chen and Pratt [1984] use the DCT on 16x16 pixel blocks, for monochrome

and NTSC color (YIQ format) images. DCT coefficients (scanned in the zig-zag manner described above) whose magnitudes exceed a given threshold are transmitted; all others are set to zero. To transmit a coefficient, the difference between the coefficient and the threshold is quantized and transmitted to the receiver using a Huffman code. A data buffer is used to provide a fixed data rate to the channel, and is constantly monitored to prevent overflow. If the buffer becomes too full, the number of bits used to quantize the DCT coefficients is reduced. Since many coefficients of each DCT block are zero, a second Huffman code is employed to perform a run-length coding for strings of zero coefficients. For color images, DCT coefficients for the Y, I, and Q components are transmitted in a similar manner. A hardware implementation of this system exists which can transmit NTSC video at 1.5 Mbits/sec. Although this system was not originally designed to operate in a progressive manner, it can be (and has been) modified to perform progressive transmission at low data rates.

Chitprasert and Rao [1990] use an 8x8 DCT system based on that of Chen and Pratt, with some modifications. The DCT coefficients are classified by their energies, and weighted by a model of the human visual system (HVS), to determine order of transmission to the receiver. Their claim is that the HVS weighting improves the perceptual quality of early coding stages, although an objective measure (SNR) indicates a slight decrease in image quality. The result is a "classification map" and a "transmission map" which must be sent as

overhead to the receiver for each frame. Quantization of the AC DCT coefficients is performed using a normalized, nonuniform Laplacian quantizer. Therefore, standard deviation matrices must also be transmitted as additional overhead to the receiver. Total overhead is claimed to be approximately 0.07 bits/pixel, with "good" quality output achieved at a total rate of 0.3-0.5 bits/pixel. Complexity of this system is somewhat high due to the DCT and classification schemes, but the system performance appears good enough to warrant further investigation.

Non-Transform Schemes

Frank, Daniels and Unangst [1980] develop a lossless "growth-geometry" scheme, which initially is used only for bilevel images. The scheme uses "seed" pixels, and "grows" portions of the images around them according to a selected rule. The growth rule is chosen from a predetermined set of rules. Encoding is essentially a pattern recognition task, to locate appropriate seed pixels and select a growth type. A small hardware implementation of the scheme exists for bilevel images. Suggestions are offered for extending the scheme to multilevel images (e.g., encoding each bit plane as a single bilevel image), but the potential for good multilevel image compression and a low complexity implementation seems small, especially for natural scenes.

Knowlton [1980] divides the source image using a binary tree structure. A large rectangular block approximation of the image is progressively halved and

refined, and eventually reproduces the original image losslessly. Operation of the system is independent of the characteristics of the source image, and the computational requirement is linear in the number of source image pixels. This should simplify a hardware implementation. The compression rate for this scheme is dependent upon the number of possible gray levels in the source image, 16 for the described research. The scheme has also been applied to bilevel images with good results. One drawback of the system is that early coding passes are more coarse than other schemes, requiring more passes to obtain a recognizable scene. In addition, good compression may be difficult for images with a much larger number of gray levels, say 256 instead of 16, as the amount of encoder data would approximately double in this case.

Sloan and Tanimoto [1979] use a pyramid structure to losslessly encode the source image. Progressive transmission is achieved by transmitting the levels of the pyramid. The lowest level of the pyramid is the NxN source image. The next pyramid level has dimension N/2xN/2, etc., and the top pyramid level is 1x1. Various methods for building and transmitting the pyramid are discussed. In one method, each successive pyramid level is formed by simply using the upper left-hand pixel of a 2x2 block from the previous level. This eliminates the need to transmit this one pixel again, and requires no computation. Then, only three of the four pixels need to be transmitted in the next pass. In another, the sum of the four pixels is transmitted instead. This also requires only three additional pixels

during each successive pass, and little computation. The resultant image is lossless in both cases. A drawback of the pyramid approach is that the entire pyramid must be built before any data can be transmitted to the receiver, resulting in a transmission delay. However, the computational load for the scheme is low, and should minimize this delay. Compression does not appear to be as good as some of the other schemes here, but there are several areas of possible improvement. For example, a Huffman coder might be useful to encode the pixels and reduce the data rate.

Dreizen [1987] develops a lossless progressive transmission scheme using a pyramid structure. In this case, the pyramid structure is used along with an information measure to identify which portions of the image to develop first. A differential predictive coder, followed by a Huffman coder, is used to update the image at the receiver. Compression for the scheme ranges from 13-37 percent for the 128x128 8-bit images used. Complexity of the scheme is low, however, the received images exhibit pronounced coding artifacts even after several coding passes.

Wang and Goldberg [1989] also use pyramid structures to represent the source image. Two pyramids are created: a mean pyramid (22 pixel averages) and a difference (error) pyramid. Most of the coding effort goes toward coding the difference pyramid, which is used to refine the output at the receiver at each stage of transmission. Coding for the difference pyramid uses a vector quantizer (LBG)

designed exclusively for each portion of the difference pyramid. Quantizer errors are remembered and recoded on later passes. As a final pass, an entropy code is used to transmit the residual errors, yielding a lossless image at the receiver. This approach yields a low data rate, but is very complex and computationally intensive. In this form, a hardware implementation of the scheme would be difficult. A possible modification to the scheme to reduce complexity might be to use a lattice VQ instead.

References

W. Chen and W.K. Pratt, "Scene Adaptive Coder," IEEE Trans. Comm., pp. 225-232, March 1984.

B. Chitprasert and K.R. Rao, "Human Visual Weighted Progressive Image Transmission," IEEE Trans. Comm., pp. 1040-44, July 1990.

H. Dreizen, "Content-Driven Progressive Transmission of Grey-Scale Images," IEEE Trans. Comm., pp. 289-296, March 1987.

E. Dubois and J.L. Moncet, "Encoding and Progressive Transmission of Still Pictures in NTSC Composite Format Using Transform Domain Methods," IEEE Trans. Comm., pp. 310-319, March 1986.

A. Frank, J.D. Daniels, and D. Unangst, "Progressive Image Transmission Using A Growth-Geometry Coding," Proc. of the IEEE, pp. 897-909, July 1980.

K. Knowlton, "Progressive Transmission of Grey-Scale and Binary Pictures by Simple, Efficient, and Lossless Encoding Schemes," Proc. of the IEEE, pp. 885-896, July 1980.

K. Ngan, "Image Display Techniques Using the Cosine Transform," IEEE Trans. Acoustics, Speech and Signal Proc., pp. 173-177, Feb. 1984.

K. Sloan and S. Tanimoto, "Progressive Refinement of Raster Images," IEEE Trans. on Computers, pp. 871-4, Nov. 1979.

L. Wang and M. Goldberg, "Progressive Transmission Using Vector Quantization on Images in Pyramid Form," IEEE Transactions on Comm., pp. 1339-1348, Dec. 1989.